COMP90042

Web search and text analysis

Workshop Week 5

Your tutor

- Winn Chow (Senior Tutor)
- winn.chow1@unimelb.edu.au
- Office: Doug McDonell - 9.23
- Here, you can find my workshop slides:
- https://github.com/winnchow/COMP90042-Workshops

Q1

1. What is **text classification**? Give some examples.
   (a) Why is text classification generally a difficult problem? What are some hurdles that need to be overcome?
   (b) Consider some (supervised) text classification problem, and discuss whether the following (supervised) machine learning models would be suitable:
      i. k-Nearest Neighbour using Euclidean distance
      ii. k-Nearest Neighbour using Cosine similarity
      iii. Decision Trees using Information Gain
      iv. Naive Bayes
      v. Logistic Regression
      vi. Support Vector Machines

Q1 Give some examples

* Topic classification
* Sentiment analysis
* Authorship attribution
* Native-language identification
* Automatic fact-checking

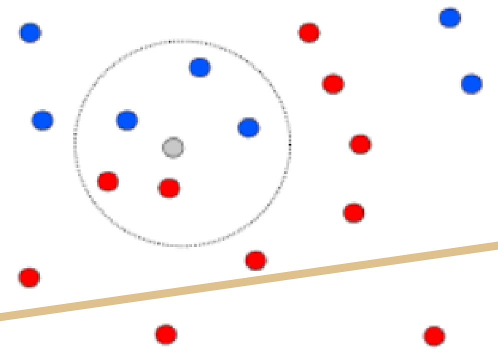
Building a Text classifier

1. Identify a task of interest
2. Collect an appropriate corpus
3. Carry out annotation
4. Select features
5. Choose a machine learning algorithm
6. Tune hyperparameters using held-out development data
7. Repeat earlier steps as needed
8. Train final model
9. Evaluate model on held-out test data

Multi-word features (e.g. bi-grams, tri-grams) suffer from a **sparse data problem**

K-Nearest Neighbour

- Classify based on majority class of *k*-nearest training examples in feature space



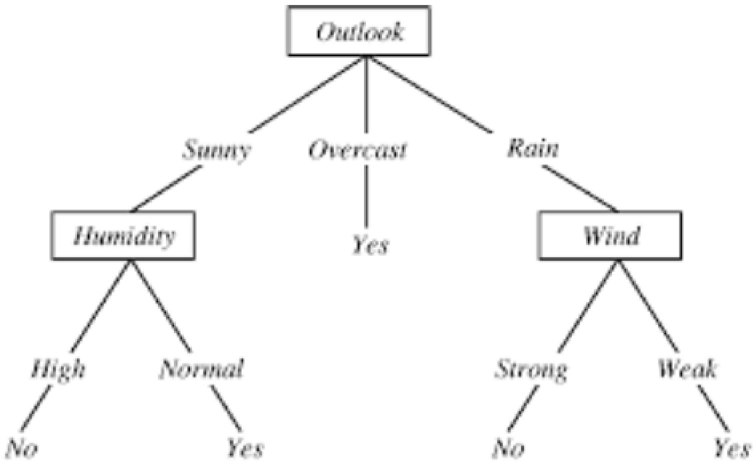
- Definition of nearest can vary
  * Euclidean distance
  * Cosine distance
- Pros: Simple, effective; no training required; inherently multiclass; optimal with infinite data
- Cons: Have to select *k*; issues with unbalanced classes; often slow (need to find those *k*-neighbours); features must be selected carefully

- Based on the number of words => document length

- Based on distribution of word types => better
- Not work well in high-dimensions

Decision tree

- Construct a tree where nodes correspond to tests on individual features
- Leaves are final class decision
- Based on greedy maximization of mutual information
- Pros: in theory, very interpretable; fast to build and test; feature representation/scaling irrelevant; good for small feature sets, handles non-linearly-separable problems
- Cons: In practice, often not that interpretable; highly redundant sub-trees; not competitive for large feature sets

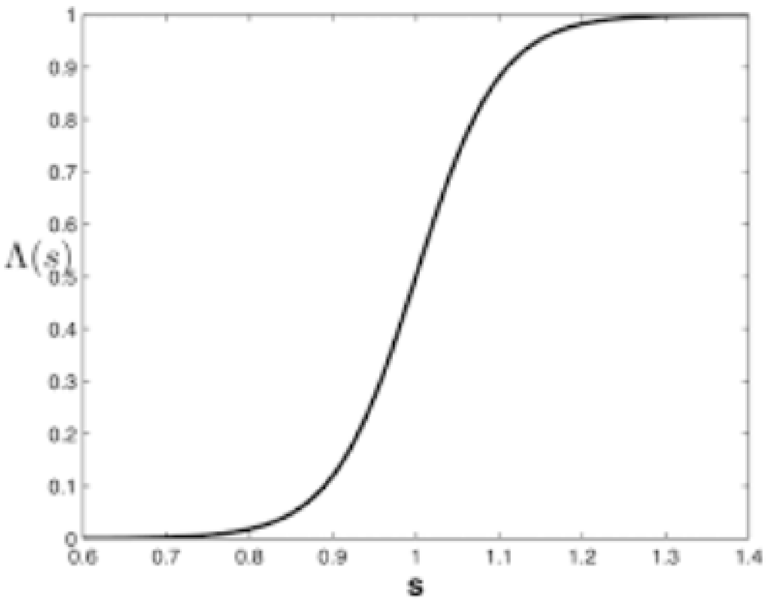


- But Information Gain (reduction of uncertainty) tends to prefer rare features

Naïve Bayes

- Pros: Fast to “train” and classify; robust, low-variance; good for low data situations; optimal classifier if independence assumption is correct; extremely simple to implement.
- Cons: Independence assumption rarely holds; low accuracy compared to similar methods in most situations; smoothing required for unseen class/feature combinations

Logistic Regression



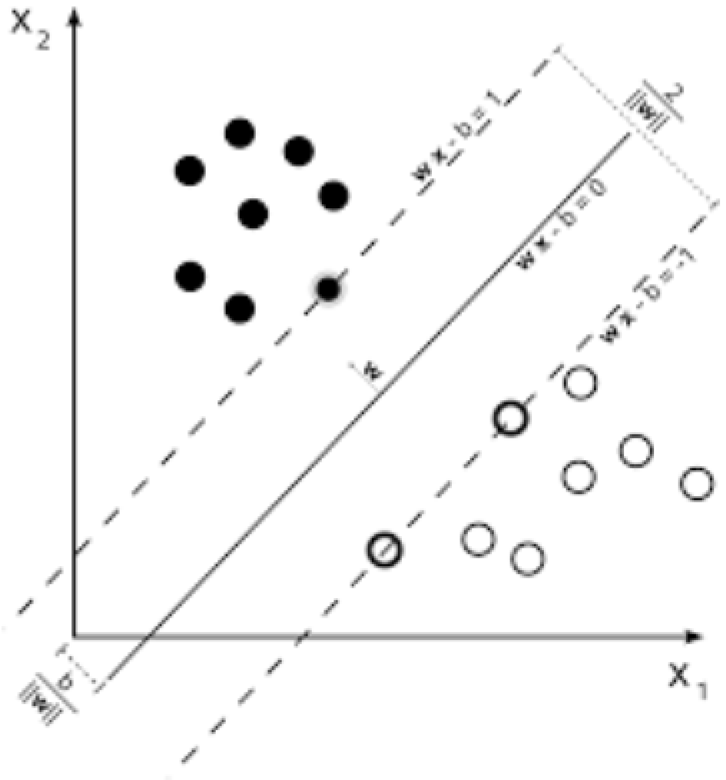
- A classifier, despite its name
- A linear model, but uses *softmax* "squashing" to get valid probability

$$p(c_n|f_1 \ldots f_m) = \frac{1}{Z} \cdot \exp\left(\sum_{i=0}^{m} w_i f_i\right)$$

- Training maximizes probability of training data subject to regularization which encourages low or sparse weights

Support vector machines

- Finds hyperplane which separates the training data with maximum margin
  - Allows for some misclassification
- Weight vector is a sum of support vectors (examples on the margin)



- Pros: fast and accurate linear classifier; can do non-linearity with kernel trick; works well with huge feature sets
- Cons: Multiclass classification awkward; feature scaling can be tricky; deals poorly with class imbalances; uninterpretable

Q2

2. For the following "corpus" of two documents:

   1. `how much wood would a wood chuck chuck if a wood chuck would chuck wood`
   2. `a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood`

   (a) Which of the following sentences: `a wood could chuck`; `wood would a chuck`; is more probable, accoding to:

   i. An unsmoothed uni-gram language model?
   ii. A uni-gram language model, with Laplacian ("add-one") smoothing?
   iii. An unsmoothed bi-gram language model?
   iv. A bi-gram language model, with Laplacian smoothing?
   v. An unsmoothed tri-gram language model?
   vi. A tri-gram language model, with Laplacian smoothing?

Uni-gram language model

When n = 1, a unigram model

$$P(w_1, w_2, .. w_m) = \prod_{i=1}^{m} P(w_i)$$

$$P(w_i) = \frac{C(w_i)}{M}$$

- M is the total number of tokens

1. how much wood would a wood chuck chuck if a wood chuck would chuck wood </s>
2. a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood </s>

Uni-gram language model

W	a	chuck	could	he	how	if	much	the	wood	would	</s>	Total (M)
Count(W)												
P(W)												

|V| = # of word types = ?

$$P(A) = P(\mathtt{a})P(\mathtt{wood})P(\mathtt{could})P(\mathtt{chuck})P(\mathtt{</s>})$$

$$P(B) = P(\mathtt{wood})P(\mathtt{would})P(\mathtt{a})P(\mathtt{chuck})P(\mathtt{</s>})$$

Uni-gram language model

W	a	chuck	could	he	how	if	much	the	wood	would	</s>	Total (M)
Count(W)	4	9	1	1	1	2	1	1	8	4	2	34
P(W)	4/34	9/34	1/34	1/34	1/34	2/34	1/34	1/34	8/34	4/34	2/34	

|V| = # of word types = 11

$$\begin{aligned} P(A) &= P(\texttt{a})P(\texttt{wood})P(\texttt{could})P(\texttt{chuck})P(\texttt{</s>}) \\ &= \frac{4}{34} \times \frac{8}{34} \times \frac{1}{34} \times \frac{9}{34} \times \frac{2}{34} \approx 1.27 \times 10^{-5} \\ P(B) &= P(\texttt{wood})P(\texttt{would})P(\texttt{a})P(\texttt{chuck})P(\texttt{</s>}) \\ &= \frac{8}{34} \times \frac{4}{34} \times \frac{4}{34} \times \frac{9}{34} \times \frac{2}{34} \approx 5.07 \times 10^{-5} \end{aligned}$$

Uni-gram language model + Laplacian ("add-one") smoothing

For unigram models (**V**= the vocabulary),

$$P_{add1}(w_i) = \frac{C(w_i) + 1}{M + |\boldsymbol{V}|}$$

|V| = # of word types = 11

W	a	chuck	could	he	how	if	much	the	wood	would	</s>	Total (M)
Count(W)	4	9	1	1	1	2	1	1	8	4	2	34
P(W)												

Uni-gram language model
+ Laplacian ("add-one") smoothing

$$
\begin{aligned}
P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a})P_{\mathrm{L}}(\texttt{wood})P_{\mathrm{L}}(\texttt{could})P_{\mathrm{L}}(\texttt{chuck})P_{\mathrm{L}}(\texttt{</s>}) \\
&= \frac{5}{45} \times \frac{9}{45} \times \frac{2}{45} \times \frac{10}{45} \times \frac{3}{45} \approx 1.46 \times 10^{-5} \\
P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood})P_{\mathrm{L}}(\texttt{would})P_{\mathrm{L}}(\texttt{a})P_{\mathrm{L}}(\texttt{chuck})P_{\mathrm{L}}(\texttt{</s>}) \\
&= \frac{9}{45} \times \frac{5}{45} \times \frac{5}{45} \times \frac{10}{45} \times \frac{3}{45} \approx 3.66 \times 10^{-5}
\end{aligned}
$$

|V| = # of word types = 11

W	a	chuck	could	he	how	if	much	the	wood	would	</s>	Total (M)
Count(W)	4	9	1	1	1	2	1	1	8	4	2	34
P(W)	5/45	10/45	2/45	2/45	2/45	3/45	2/45	2/45	9/45	5/45	3/45	

Bi-gram language model

When n = 2, a bigram model

$$P(w_1, w_2, .. w_m) = \prod_{i=1}^{m} P(w_i | w_{i-1})$$

$$P(w_i | w_{i-1}) = \frac{C(w_{i-1} w_i)}{C(w_{i-1})}$$

1. <s> how much wood would a wood chuck chuck if a wood chuck would chuck wood </s>
2. <s> a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood </s>

Bi-gram language model

W_1 W_2	<s> a	a wood	wood could	could chuck	chuck </s>	<s> wood	wood would	would a	a chuck	chuck </s>
Count (W_1 W_2)										
Count (W_1)										
P(W_2|W_1)										

$$P(A) = P(\texttt{a}|\texttt{<s>})P(\texttt{wood}|\texttt{a})P(\texttt{could}|\texttt{wood})P(\texttt{chuck}|\texttt{could})P(\texttt{</s>}|\texttt{chuck})$$

$$P(B) = P(\texttt{wood}|\texttt{<s>})P(\texttt{would}|\texttt{wood})P(\texttt{a}|\texttt{would})P(\texttt{chuck}|\texttt{a})P(\texttt{</s>}|\texttt{chuck})$$

Bi-gram language model

W_1 W_2	<s> a	a wood	wood could	could chuck	chuck </s>	<s> wood	wood would	would a	a chuck	chuck </s>
Count (W_1 W_2)	1	4	0	1	0	0	1	1	0	0
Count (W_1)	2	4	8	1	9	2	8	4	4	9
P(W_2|W_1)	1/2	4/4	0/8	1/1	0/9	0/2	1/8	1/4	0/4	0/9

$$\begin{aligned}
P(A) &= P(\texttt{a}|\texttt{<s>})P(\texttt{wood}|\texttt{a})P(\texttt{could}|\texttt{wood})P(\texttt{chuck}|\texttt{could})P(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{1}{2} \times \frac{4}{4} \times \frac{0}{8} \times \frac{1}{1} \times \frac{0}{9} = 0 \\
P(B) &= P(\texttt{wood}|\texttt{<s>})P(\texttt{would}|\texttt{wood})P(\texttt{a}|\texttt{would})P(\texttt{chuck}|\texttt{a})P(\texttt{</s>}|\texttt{chuck}) \\
&= \frac{0}{2} \times \frac{1}{8} \times \frac{1}{4} \times \frac{0}{4} \times \frac{0}{9} = 0
\end{aligned}$$

Bi-gram language model + Laplacian (“add-one”) smoothing

For bigram models,

$$P_{add1}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) + 1}{C(w_{i-1}) + |V|}$$

|V| = # of word types = ?
Note that <s> is not included in V as it is never generated.

W_1 W_2	<s> a	a wood	wood could	could chuck	chuck </s>	<s> wood	wood would	would a	a chuck	chuck </s>
Count (W_1 W_2)	1	4	0	1	0	0	1	1	0	0
Count (W_1)	2	4	8	1	9	2	8	4	4	9
P(W_2|W_1)										

Bi-gram language model
+ Laplacian ("add-one") smoothing

$$
\begin{aligned}
P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a}|\texttt{<s>})P_{\mathrm{L}}(\texttt{wood}|\texttt{a})P_{\mathrm{L}}(\texttt{could}|\texttt{wood})P_{\mathrm{L}}(\texttt{chuck}|\texttt{could})P_{\mathrm{L}}(\texttt{</s>}|\texttt{chuc} \\
&= \frac{2}{13} \times \frac{5}{15} \times \frac{1}{19} \times \frac{2}{12} \times \frac{1}{20} \approx 2.25 \times 10^{-5} \\
P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood}|\texttt{<s>})P_{\mathrm{L}}(\texttt{would}|\texttt{wood})P_{\mathrm{L}}(\texttt{a}|\texttt{would})P_{\mathrm{L}}(\texttt{chuck}|\texttt{a})P_{\mathrm{L}}(\texttt{</s>}|\texttt{chuc} \\
&= \frac{1}{13} \times \frac{2}{19} \times \frac{2}{15} \times \frac{1}{15} \times \frac{1}{20} \approx 3.60 \times 10^{-6}
\end{aligned}
$$

|V| = # of word types = 11
Note that <s> is not included in V as it is never generated.

W_1 W_2	<s> a	a wood	wood could	could chuck	chuck </s>	<s> wood	wood would	would a	a chuck	chuck </s>
Count (W_1 W_2)	1	4	0	1	0	0	1	1	0	0
Count (W_1)	2	4	8	1	9	2	8	4	4	9
$P(W_2 \mid W_1)$	2/13	5/15	1/19	2/12	1/20	1/13	2/19	2/15	1/15	1/20

Tri-gram language model

When n = 3, a trigram model

$$P(w_1, w_2, .. w_m) = \prod_{i=1}^{m} P(w_i | w_{i-2} w_{i-1})$$

$$P(w_i | w_{i-2} w_{i-1}) = \frac{C(w_{i-2} w_{i-1} w_i)}{C(w_{i-2} w_{i-1})}$$

1. <s><s> how much wood would a wood chuck chuck if a wood chuck would chuck wood </s>
2. <s><s> a wood chuck would chuck the wood he could chuck if a wood chuck would chuck wood </s>

Tri-gram language model

$W_1\ W_2\ W_3$	<s> <s> a	<s> a wood	a wood could	wood could chuck	could chuck </s>	<s> <s> wood	<s> wood would	wood would a	would a chuck	a chuck </s>
Count ($W_1\ W_2\ W_3$)										
Count ($W_1\ W_2$)										
P($W_3 \mid W_1\ W_2$)										

$$P(A) = P(\texttt{a}|\texttt{<s> <s>})P(\texttt{wood}|\texttt{<s> a})\cdots P(\texttt{</s>}|\texttt{could chuck})$$

$$P(B) = P(\texttt{wood}|\texttt{<s> <s>})P(\texttt{would}|\texttt{<s> wood})\cdots P(\texttt{</s>}|\texttt{a chuck})$$

Tri-gram language model

$W_1\ W_2\ W_3$	<s> <s> a	<s> a wood	a wood could	wood could chuck	could chuck </s>	<s> <s> wood	<s> wood would	wood would a	would a chuck	a chuck </s>
Count ($W_1\ W_2\ W_3$)	1	1	0	0	0	0	0	1	0	0
Count ($W_1\ W_2$)	2	1	4	0	1	2	0	1	1	0
$P(W_3 \mid W_1\ W_2)$	1/2	1/1	0/4	0/0	0/1	0/2	0/0	1/1	0/1	0/0

$$
\begin{aligned}
P(A) &= P(\texttt{a}|\texttt{<s> <s>})P(\texttt{wood}|\texttt{<s> a})\cdots P(\texttt{</s>}|\texttt{could chuck}) \\
&= \frac{1}{2}\times\frac{1}{1}\times\frac{0}{4}\times\frac{0}{0}\times\frac{0}{1} = ? \\
P(B) &= P(\texttt{wood}|\texttt{<s> <s>})P(\texttt{would}|\texttt{<s> wood})\cdots P(\texttt{</s>}|\texttt{a chuck}) \\
&= \frac{0}{2}\times\frac{0}{0}\times\frac{1}{1}\times\frac{0}{1}\times\frac{0}{0} = ?
\end{aligned}
$$

Tri-gram language model
+ Laplacian (“add-one”) smoothing

|V| = # of word types = ?
Note that <s> is not included in V as it is never generated.

$$P_{add1}(w_i|w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i) + 1}{C(w_{i-2}w_{i-1}) + |V|}$$

$W_1\ W_2\ W_3$	<s> <s> a	<s> a wood	a wood could	wood could chuck	could chuck </s>	<s> <s> wood	<s> wood would	wood would a	would a chuck	a chuck </s>
Count ($W_1\ W_2\ W_3$)	1	1	0	0	0	0	0	1	0	0
Count ($W_1\ W_2$)	2	1	4	0	1	2	0	1	1	0
$P(W_3 \mid W_1\ W_2)$										

Tri-gram language model + Laplacian ("add-one") smoothing

$$
\begin{aligned}
P_{\mathrm{L}}(A) &= P_{\mathrm{L}}(\texttt{a}|\texttt{<s> <s>})P_{\mathrm{L}}(\texttt{wood}|\texttt{<s> a})\cdots P_{\mathrm{L}}(\texttt{</s>}|\texttt{could chuck}) \\
&= \frac{2}{13}\times\frac{2}{12}\times\frac{1}{15}\times\frac{1}{11}\times\frac{1}{12}\approx 1.30\times 10^{-5} \\
P_{\mathrm{L}}(B) &= P_{\mathrm{L}}(\texttt{wood}|\texttt{<s> <s>})P_{\mathrm{L}}(\texttt{would}|\texttt{<s> wood})\cdots P_{\mathrm{L}}(\texttt{</s>}|\texttt{a chuck}) \\
&= \frac{1}{13}\times\frac{1}{11}\times\frac{2}{12}\times\frac{1}{12}\times\frac{1}{11}\approx 8.83\times 10^{-6}
\end{aligned}
$$

|V| = # of word types = 11
Note that <s> is not included in V as it is never generated.

$W_1\ W_2\ W_3$	<s> <s> a	<s> a wood	a wood could	wood could chuck	could chuck </s>	<s> <s> wood	<s> wood would	wood would a	would a chuck	a chuck </s>
Count ($W_1\ W_2\ W_3$)	1	1	0	0	0	0	0	1	0	0
Count ($W_1\ W_2$)	2	1	4	0	1	2	0	1	1	0
$P(W_3 \mid W_1\ W_2)$	2/13	2/12	1/15	1/11	1/12	1/13	1/11	2/12	1/12	1/11

Back-off and Interpolation

- Back–off is a smoothing strategy, where we incorporate lower–order n-gram models (in particular, for unseen contexts).
- Interpolation is a similar idea, but instead of only “falling back” to lower–order n-gram models for unseen events, we can instead consider every probability as a linear combination of all of the relevant n-gram models, where the weights are once more chosen to ensure that the probabilities of all events, given some context, sum to 1.